

NNN	NNN	CCCCCCCCCCCC	PPPPPPPPPPPPP
NNN	NNN	CCCCCCCCCCCC	PPPPPPPPPPPPP
NNN	NNN	CCCCCCCCCCCC	PPPPPPPPPPPPP
NNN	NNN	CCC	PPP PPP
NNN	NNN	CCC	PPP PPP
NNN	NNN	CCC	PPP PPP
NNNNNN	NNN	CCC	PPP PPP
NNNNNN	NNN	CCC	PPP PPP
NNNNNN	NNN	CCC	PPP PPP
NNN	NNN	NNN CCC	PPPPPPPPPPPPP
NNN	NNN	NNN CCC	PPPPPPPPPPPPP
NNN	NNN	NNN CCC	PPPPPPPPPPPPP
NNN	NNNNNN	CCC	PPP
NNN	NNNNNN	CCC	PPP
NNN	NNNNNN	CCC	PPP
NNN	NNN	CCC	PPP
NNN	NNN	CCC	PPP
NNN	NNN	CCC	PPP
NNN	NNN	CCCCCCCCCCCC	PPP
NNN	NNN	CCCCCCCCCCCC	PPP
NNN	NNN	CCCCCCCCCCCC	PPP

FILE ID**NCPSTACLP

N 8

NC
VO

```
1 0001 0 XTITLE 'Clear/Purge Parse States and Data'
2 0002 0 MODULE NCPSTACLP (IDENT = 'V04-000', LIST(NOOBJECT)) =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 ****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 ****
28 0028 1 *
29 0029 1 *
30 0030 1 **+
31 0031 1 FACILITY: Network Control Program (NCP)
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 States and data for the parsing of NCP clear and purge commands
36 0036 1
37 0037 1 ENVIRONMENT: VAX/VMS Operating System
38 0038 1
39 0039 1 AUTHOR: Darrell Duffy . CREATION DATE: 14-September-79
40 0040 1
41 0041 1 MODIFIED BY:
42 0042 1
43 0043 1 V03-019 PRD0053 Paul R. DeStefano 05-Feb-1984
44 0044 1 Allow OBJECT parameter to accept both name and number.
45 0045 1
46 0046 1 V03-018 PRD0052 Paul R. DeStefano 05-Feb-1984
47 0047 1 Complete addition of and enable X25-Access parsing.
48 0048 1
49 0049 1 V03-017 RPG0017 Bob Grosso 01-Oct-1982
50 0050 1 Clear module Configurator.
51 0051 1
52 0052 1 V03-016 RPG0016 Bob Grosso 21-Sep-1982
53 0053 1 Add clear DTE and Clear Group and Clear Tracepoint
54 0054 1
55 0055 1 V03-015 RPG0015 Bob Grosso 09-Aug-1982
56 0056 1 Add CLEAR LINE MODE/MICROCODE.
57 0057 1
```

: 58 0058 1 | V03-014 RPG0014 Bob Grosso 15-Jul-1982
: 59 0059 1 | Add CLEAR/PURGE X25-Trace, Module Configurator, Console
: 60 0060 1 | loader, looper.
: 61 0061 1 |
: 62 0062 1 | V013 RPG0013 Bob Grosso 09-Jul-1982
: 63 0063 1 | Add NI node parameters.
: 64 0064 1 | Remove CLEAR/PURGE NODE/EXECUTOR since parse tables have
: 65 0065 1 | have exceeded keyword limit.
: 66 0066 1 |
: 67 0067 1 | V012 RPG0012 Bob Grosso 10-May-1982
: 68 0068 1 | Add support for circuit MRT and RPR for NI support.
: 69 0069 1 | Change MAX BLOCK to MAX DATA.
: 70 0070 1 |
: 71 0071 1 | V011 TMH0011 Tim Halvorsen 08-Mar-1982
: 72 0072 1 | Remove prompting for CLEAR LINE, CIRCUIT and NODE
: 73 0073 1 | parameters, since it is somewhat confusing and
: 74 0074 1 | these commands are normally used only to clear the
: 75 0075 1 | entire database entry, rather than a single parameter.
: 76 0076 1 | It also makes it easier to hide latent support of
: 77 0077 1 | parameters.
: 78 0078 1 |
: 79 0079 1 | V010 TMH0010 Tim Halvorsen 20-Jan-1982
: 80 0080 1 | Add CIRCUIT TRANSPORT TYPE, VERIFICATION parameters.
: 81 0081 1 | Remove prompting of X25 specific parameters.
: 82 0082 1 |
: 83 0083 1 | V009 TMH0009 Tim Halvorsen 08-Jan-1982
: 84 0084 1 | Remove TMH0004, thus restoring RETRANSMIT TIMER
: 85 0085 1 | to a line parameter, which is what NM V3.0 finally
: 86 0086 1 | came up with.
: 87 0087 1 | Add additional prompting for CLEAR LINE.
: 88 0088 1 |
: 89 0089 1 | V008 TMH0008 Tim Halvorsen 18-Dec-1981
: 90 0090 1 | Add CLEAR OBJECT PROXY, EXECUTOR DEFAULT PROXY & NODE PROXY.
: 91 0091 1 |
: 92 0092 1 | V007 TMH0007 Tim Halvorsen 11-Nov-1981
: 93 0093 1 | Add circuit source type to clear logging.
: 94 0094 1 |
: 95 0095 1 | V006 TMH0006 Tim Halvorsen 22-Oct-1981
: 96 0096 1 | Fix CLEAR CIRCUIT TRIBUTARY to send the right
: 97 0097 1 | NICE parameter code. Add CLEAR EXEC DEFAULT ACCESS
: 98 0098 1 | and CLEAR NODE ACCESS.
: 99 0099 1 |
: 100 0100 1 | V005 TMH0005 Tim Halvorsen 18-Aug-1981
: 101 0101 1 | Change NODE MAXIMUM LINES to MAXIMUM CIRCUITS.
: 102 0102 1 | Change NODE SERVICE LINE to SERVICE CIRCUIT.
: 103 0103 1 | Change NODE LINE (loop line) to CIRCUIT.
: 104 0104 1 | Add CLEAR EXECUTOR PIPELINE QUOTA.
: 105 0105 1 |
: 106 0106 1 | V004 TMH0004 Tim Halvorsen 05-Aug-1981
: 107 0107 1 | Change RETRANSMIT TIMER to a circuit parameter
: 108 0108 1 | from a line parameter.
: 109 0109 1 |
: 110 0110 1 | V003 TMH0003 Tim Halvorsen 07-Jul-1981
: 111 0111 1 | Rename circuit maximum blocks to maximum transmits.
: 112 0112 1 | Add line clock parameter.
: 113 0113 1 | Add dispatching for CLEAR X25-xxx.
: 114 0114 1 | Add node subaddresses parameter.

NCPSTACLP
V04-000

Clear/Purge Parse States and Data

D 9
16-Sep-1984 00:05:04
14-Sep-1984 12:48:17 VAX-11 Bliss-32 V4.0-742
[NCP.SRC]NCPSTACLP.B32;1

Page 3
(1)

: 115 0115 1 :
: 116 0116 1 :
: 117 0117 1 :
: 118 0118 1 :
: 119 0119 1 :
: 120 0120 1 :
: 121 0121 1 :--

V002 TMH0002 Tim Halvorsen 11-Jun-1981
Add CLEAR CIRCUIT
V001 LMK0001 Len Kawell 5-Jan-1981
Remove HOST ID prompt from CLEAR EXECUTOR.

NCF
V04

```
123      0122 1 %SBTTL 'Definitions'  
124      0123 1  
125      0124 1  
126      0125 1 ! INCLUDE FILES:  
127      0126 1 !  
128      0127 1  
129      0128 1     LIBRARY 'LIBS:NMALIBRY';  
130      0129 1     LIBRARY 'LIBS:NCPLIBRY';  
131      0130 1     LIBRARY 'SYSSLIBRARY:TPAMAC';  
132      0131 1  
133      0132 1 ! EXTERNAL REFERENCES:  
134      0133 1 !  
135      0134 1 !  
136      0135 1 !  
137      0136 1 EXTERNAL  
138      0137 1     NCPNG_STTBL_CNO.          ! Clear NODE  
139      0138 1     NCPNG_KYTBL_CNO.        ! Clear EXECUTOR  
140      0139 1     NCPNG_STTBL_CEX.        ! Clear Configurator  
141      0140 1     NCPNG_KYTBL_CEX.  
142      0141 1     NCPNG_STTBL_CCF.        ! Clear Console  
143      0142 1     NCPNG_KYTBL_CCF.  
144      0143 1     NCPNG_STTBL_CCS.        ! Clear Loader  
145      0144 1     NCPNG_KYTBL_CCS.  
146      0145 1     NCPNG_STTBL_CLD.        ! Clear Looper  
147      0146 1     NCPNG_KYTBL_CLD.  
148      0147 1     NCPNG_STTBL_CLP.        ! Clear X25-ACCESS  
149      0148 1     NCPNG_KYTBL_CLP.  
150      0149 1     NCPNG_STTBL_CAC.        ! Clear X25-PROTOCOL  
151      0150 1     NCPNG_KYTBL_CAC.  
152      0151 1     NCPNG_STTBL_CPR.        ! Clear X25-PROTOCOL Group  
153      0152 1     NCPNG_KYTBL_CPR.  
154      0153 1     NCPNG_STTBL_CPRDTE.      ! Clear X25-PROTOCOL DTE  
155      0154 1     NCPNG_KYTBL_CPRDTE.  
156      0155 1     NCPNG_STTBL_CPRGRP.      ! Clear X25-SERVER  
157      0156 1     NCPNG_KYTBL_CPRGRP.  
158      0157 1     NCPNG_STTBL_CSE.        ! Clear X25-TRACE  
159      0158 1     NCPNG_KYTBL_CSE.  
160      0159 1     NCPNG_STTBL_CTR.  
161      0160 1     NCPNG_KYTBL_CTR.  
162      0161 1     NCPNG_STTBL_CTRPT.  
163      0162 1     NCPNG_KYTBL_CTRPT.  
164      0163 1     NCPNG_STTBL_C9S.  
165      0164 1     NCPNG_KYTBL_C9S.  
166      0165 1     PBKSG_CNO_ENT;  
167      0166 1  
168      0167 1     ACT_DFN           ! External symbols for action routines  
169      0168 1  
170      0169 1 EXTERNAL ROUTINE  
171      0170 1     ACTSNXT_STATE;       ! Set vector to next state table
```

```
: 173      0171 1 %SBTTL 'Parameter blocks'  
.: 174      0172 1  
.: 175      0173 1 :  
.: 176      0174 1 : BIND DATA:  
.: 177      0175 1 :  
.: 178      0176 1 :  
.: 179      0177 1 :  
.: 180      0178 1 : Parameter Blocks  
.: 181      0179 1 :  
.: 182      0180 1 :  
.: 183      0181 1 :  
.: 184      0182 1 : General blocks  
.: 185      0183 1 :  
.: 186      0184 1 :  
.: 187      P 0185 1 : BUILD_PBK  
.: 188      P 0186 1 :  
.: 189      P 0187 1 : (CLPU,  
.: 190      P 0188 1 :  
.: 191      P 0189 1 : ALL, LITB, 0, VRB_ALL,  
.: 192      P 0190 1 : KWN, LITB, NMASC_ENT_KNO, VRB_ENT,  
.: 193      P 0191 1 :  
.: 194      0192 1 : )
```

196 0193 1
197 0194 1
198 0195 1
199 0196 1
200 0197 1
201 P 0198 1 BUILD_PCL
202 P 0199 1 (CCI,
203 P 0200 1
204 P 0201 1 STA, LITB, PCCI-STA,
205 P 0202 1 SER, LITB, PCCI-SER,
206 P 0203 1 CTM, LITB, PCCI-LCT,
207 P 0204 1 COS, LITB, PCCI-COS,
208 P 0205 1 MRT, LITB, PCCI-MRT,
209 P 0206 1 RPR, LITB, PCCI-RPR,
210 P 0207 1 HET, LITB, PCCI-HET,
211 P 0208 1 LIT, LITB, PCCI-LIT,
212 P 0209 1 BLK, LITB, PCCI-BLK,
213 P 0210 1 MRC, LITB, PCCI-MRC,
214 P 0211 1 RCT, LITB, PCCI-RCT,
215 P 0212 1 NUM, LITB, PCCI-NUM,
216 P 0213 1 POL, LITB, PCCI-POL,
217 P 0214 1 OWN, LITB, PCCI-OWN,
218 P 0215 1 LIN, LITB, PCCI-LIN,
219 P 0216 1 USE, LITB, PCCI-USE,
220 P 0217 1 TYP, LITB, PCCI-TYP,
221 P 0218 1 DTE, LITB, PCCI-DTE,
222 P 0219 1 CHN, LITB, PCCI-CHN,
223 P 0220 1 MBL, LITB, PCCI-MBL,
224 P 0221 1 MWI, LITB, PCCI-MWI,
225 P 0222 1 TRI, LITB, PCCI-TRI,
226 P 0223 1 BBT, LITB, PCCI-BBT,
227 P 0224 1 TRT, LITB, PCCI-TRT,
228 P 0225 1 MRB, LITB, PCCI-MRB,
229 P 0226 1 MTR, LITB, PCCI-MTR,
230 P 0227 1 ACB, LITB, PCCI-ACB,
231 P 0228 1 ACI, LITB, PCCI-ACI,
232 P 0229 1 IAB, LITB, PCCI-IAB,
233 P 0230 1 IAI, LITB, PCCI-IAI,
234 P 0231 1 IAT, LITB, PCCI-IAT,
235 P 0232 1 DYB, LITB, PCCI-DYB,
236 P 0233 1 DYI, LITB, PCCI-DYI,
237 P 0234 1 DYT, LITB, PCCI-DYT,
238 P 0235 1 DTH, LITB, PCCI-DTH,
239 P 0236 1 VER, LITB, PCCI-VER,
240 P 0237 1 XPT, LITB, PCCI-XPT,
241 P 0238 1
242 P 0239 1 . END...
243 P 0240 1
244 P 0241 1)
245 P 0242 1
246 P 0243 1
247 P 0244 1 BUILD_PBK
248 P 0245 1 (CCI,
249 P 0246 1
250 P 0247 1
251 P 0248 1 ALL, LITB, 0, VRB_ALL.
252 P 0249 1 ENT, TKN, , VRB_ENT,

253 P 0250 1
254 P 0251 1 STA, LITB, 0, .
255 P 0252 1 SER, LITB, 0, .
256 P 0253 1 CTM, LITB, 0, .
257 P 0254 1 COS, LITB, 0, .
258 P 0255 1 MRT, LITB, 0, .
259 P 0256 1 RPR, LITB, 0, .
260 P 0257 1 HET, LITB, 0, .
261 P 0258 1 LIT, LITB, 0, .
262 P 0259 1 BLK, LITB, 0, .
263 P 0260 1 MRC, LITB, 0, .
264 P 0261 1 RCT, LITB, 0, .
265 P 0262 1 NUM, LITB, 0, .
266 P 0263 1 POL, LITB, 0, .
267 P 0264 1 OWN, LITB, 0, .
268 P 0265 1 LIN, LITB, 0, .
269 P 0266 1 USE, LITB, 0, .
270 P 0267 1 TYP, LITB, 0, .
271 P 0268 1 DTE, LITB, 0, .
272 P 0269 1 CHN, LITB, 0, .
273 P 0270 1 MBL, LITB, 0, .
274 P 0271 1 MWI, LITB, 0, .
275 P 0272 1 TRI, LITB, 0, .
276 P 0273 1 BBT, LITB, 0, .
277 P 0274 1 TRT, LITB, 0, .
278 P 0275 1 MRB, LITB, 0, .
279 P 0276 1 MTR, LITB, 0, .
280 P 0277 1 ACB, LITB, 0, .
281 P 0278 1 ACI, LITB, 0, .
282 P 0279 1 IAB, LITB, 0, .
283 P 0280 1 IAI, LITB, 0, .
284 P 0281 1 IAT, LITB, 0, .
285 P 0282 1 DYB, LITB, 0, .
286 P 0283 1 DYI, LITB, 0, .
287 P 0284 1 DYT, LITB, 0, .
288 P 0285 1 DTH, LITB, 0, .
289 P 0286 1
290 P 0287 1 VER, LITB, 0, .
291 P 0288 1 XPT, LITB, 0, .
292 P 0289 1)
293 P 0290 1
294 P 0291 1
295 P 0292 1 BUILD_SDB
296 P 0293 1
297 P 0294 1 (CCI, NMASC_ENT_CIR, VRB_ENT, CCI)

299 0295 1
300 0296 1
301 0297 1
302 0298 1
303 0299 1
304 P 0300 1 BUILD_PCL
305 P 0301 1 (CLI,
306 P 0302 1
307 P 0303 1
308 P 0304 1 STA, LITB, PCLI_STA, .
309 P 0305 1 SVM, LITB, PCLI_SER, .
310 P 0306 1 CTM, LITB, PCLI_LCT, .
311 P 0307 1 COS, LITB, PCLI_COS, .
312 P 0308 1 DEV, LITB, PCLI_DEV, .
313 P 0309 1 PRO, LITB, PCLI_PRO, .
314 P 0310 1 DPX, LITB, PCLI_DUP, .
315 P 0311 1 CON, LITB, PCLI_CON, .
316 P 0312 1 CLO, LITB, PCLI_CLO, .
317 P 0313 1 TYP, LITB, PCLI_LTY, .
318 P 0314 1 STM, LITB, PCLI_STI, .
319 P 0315 1 NTM, LITB, PCLI_NTI, .
320 P 0316 1 HTI, LITB, PCLIHTI, .
321 P 0317 1 MBL, LITB, PCLI_MBL, .
322 P 0318 1 MRT, LITB, PCLI_MRT, .
323 P 0319 1 MWI, LITB, PCLI_MWI, .
324 P 0320 1 TRB, LITB, PCLI_TRI, .
325 P 0321 1 SLT, LITB, PCLI_SLT, .
326 P 0322 1 DDT, LITB, PCLI_DDT, .
327 P 0323 1 DLT, LITB, PCLI_DLT, .
328 P 0324 1 SRT, LITB, PCLI_SRT, .
329 P 0325 1 BFN, LITB, PCLI_BFN, .
330 P 0326 1
331 P 0327 1 MCD, LITB, PCLI_MCD, .
332 P 0328 1 XMD, LITB, PCLI_XMD, .
333 P 0329 1
334 P 0330 1 , END, . . .
335 P 0331 1)
336 P 0332 1
337 P 0333 1
338 P 0334 1 BUILD_PBK
339 P 0335 1 (CLI,
340 P 0336 1
341 P 0337 1
342 P 0338 1 ALL, LITB, 0, VRB_ALL,
343 P 0339 1 ENT, TKN, , VRB_ENT,
344 P 0340 1
345 P 0341 1 STA, LITB, 0, .
346 P 0342 1 SVM, LITB, 0, .
347 P 0343 1 CTM, LITB, 0, .
348 P 0344 1 COS, LITB, 0, .
349 P 0345 1 DEV, LITB, 0, .
350 P 0346 1 PRO, LITB, 0, .
351 P 0347 1 DPX, LITB, 0, .
352 P 0348 1 CON, LITB, 0, .
353 P 0349 1 CLO, LITB, 0, .
354 P 0350 1 TYP, LITB, 0, .
355 P 0351 1 STM, LITB, 0, .

```
: 356      P 0352 1      NTM, LITB, 0, .
: 357      P 0353 1      HTI, LITB, 0, .
: 358      P 0354 1      MBL, LITB, 0, .
: 359      P 0355 1      MRT, LITB, 0, .
: 360      P 0356 1      MWI, LITB, 0, .
: 361      P 0357 1      TRB, LITB, 0, .
: 362      P 0358 1      SLT, LITB, 0, .
: 363      P 0359 1      DDT, LITB, 0, .
: 364      P 0360 1      DLT, LITB, 0, .
: 365      P 0361 1      SRT, LITB, 0, .
: 366      P 0362 1      BFN, LITB, 0, .
: 367      P 0363 1
: 368      P 0364 1      MCD, LITB, 0, .
: 369      P 0365 1      XMD, LITB, 0, .
: 370      P 0366 1
: 371      0367 1      )
: 372      0368 1
: 373      P 0369 1      BUILD_SDB
: 374      P 0370 1
: 375      0371 1      (CLI, NMASC_ENT_LIN, VRB_ENT, CLI)
```

: 377 0372 1
: 378 0373 1
: 379 0374 1 Logging
: 380 0375 1
: 381 0376 1
: 382 P 0377 1 BUILD_PCL
: 383 P 0378 1 (CLO,
: 384 P 0379 1
: 385 P 0380 1
: 386 P 0381 1 NAM, LITB. PCLO_LNA, .
: 387 P 0382 1 SNO, NADR. PCLO_SIN, .
: 388 P 0383 1 EVL, ESET. PCLO_EVE, VRB_EVE.
: 389 P 0384 1 . END, . . .
: 390 P 0385 1
: 391 P 0386 1)
: 392 0387 1
: 393 0388 1
: 394 P 0389 1 BUILD_PBK
: 395 P 0390 1 (CLO,
: 396 P 0391 1
: 397 P 0392 1
: 398 P 0393 1 ALL, LITB, 0, VRB_ALL.
: 399 P 0394 1 EVK, LITB, 0, CLO_EVL.
: 400 P 0395 1 KNE, LITB, 0, CLO_EVL.
: 401 P 0396 1 EVL, LITB, 0, CLO_SNO.
: 402 P 0397 1 SEX, LITL, 0, CLO_SNO.
: 403 P 0398 1 SNO, NADR,
: 404 P 0399 1 NAM, LITB, 0,
: 405 P 0400 1 CIR, ESCI, . VRB_EVE,
: 406 P 0401 1 LIN, ESLI, . VRB_EVE,
: 407 P 0402 1 NOD, ESNO, . VRB_EVE,
: 408 P 0403 1
: 409 0404 1)
: 410 0405 1
: 411 P 0406 1 BUILD_SDB
: 412 P 0407 1 (CLO, NMASC_ENT_LOG, VRB_ENT, (CLO)
: 413 0408 1

: 415 0409 1
: 416 0410 1
: 417 0411 1 : Objects
: 418 0412 1
: 419 0413 1
: 420 P 0414 1 BUILD_PCL
: 421 P 0415 1
: 422 P 0416 1 (COB,
: 423 P 0417 1
: 424 P 0418 1 NUM, LITB, PCOB_NUM, .
: 425 P 0419 1 FIL, LITB, PCOB_FID, .
: 426 P 0420 1 PRV, LITB, PCOB_PRV, .
: 427 P 0421 1 USR, LITB, PCOB_USR, .
: 428 P 0422 1 ACC, LITB, PCOB_ACC, .
: 429 P 0423 1 PSW, LITB, PCOB_PSW, .
: 430 P 0424 1 PRX, LITB, PCOB_PRX, .
: 431 P 0425 1
: 432 P 0426 1 . END. . .
: 433 P 0427 1)
: 434 0428 1
: 435 0429 1
: 436 P 0430 1 BUILD_PBK
: 437 P 0431 1
: 438 P 0432 1 (COB,
: 439 P 0433 1
: 440 P 0434 1 ENT, TKN, . VRB_ENT,
: 441 P 0435 1
: 442 P 0436 1 ALL, LITB, 0, VRB_ALL.
: 443 P 0437 1 NUM, LITB, 0, .
: 444 P 0438 1 FIL, LITB, 0, .
: 445 P 0439 1 PRV, LITB, 0, .
: 446 P 0440 1 USR, LITB, 0, .
: 447 P 0441 1 ACC, LITB, 0, .
: 448 P 0442 1 PSW, LITB, 0, .
: 449 P 0443 1 PRX, LITB, 0, .
: 450 P 0444 1)
: 451 0445 1
: 452 0446 1
: 453 P 0447 1 BUILD_SDB
: 454 P 0448 1
: 455 0449 1 (COB, -NMASC_SENT_OBJ, VRB_ENT, COB)

```
: 457      0450 1 %SBTTL 'Prompt strings'  
: 458      0451 1  
: 459      0452 1 :  
: 460      0453 1 : Build prompt strings  
: 461      0454 1 !  
: 462      0455 1 :  
: 463      0456 1 BIND  
: 464      0457 1 :  
: 465      P 0458 1 PROMPT_STRINGS  
: 466      (CLPU,  
: 467      P 0460 1 :  
: 468      L 0461 1 ENT, %STRING('CIRCUIT, EXECUTOR KNOWN LINE:', CRLF,  
: 469      P 0462 1 LOGGING, MODULE NODE OBJECT: ', CRLF,  
: 470      P 0463 1 MOD, %STRING(' X25-ACCESS, X25-PROTOCOL, X25-SERVER,', CRLF,  
: 471      P 0464 1 !!! X25-TRACE, X29-SERVER): ', CRLF,  
: 472      L 0465 1 MOD, %STRING('Module CONFIGURATOR, CONSOLE, LOADER,', CRLF,  
: 473      L 0466 1 LOOPER, X25-ACCESS, X25-PROTOCOL, ', CRLF,  
: 474      P 0467 1 X25-SERVER, X25-TRACE, X29-SERVER): ',  
: 475      P 0468 1 KWN, '(LINES, LOGGING, NODES, OBJECTS): ',  
: 476      P 0469 1 :  
: 477      0470 2 )  
: 478      0471 1 :  
: 479      0472 1 :  
: 480      0473 1 :  
: 481      P 0474 1 PROMPT_STRINGS  
: 482      (CNO,  
: 483      P 0476 1 :  
: 484      P 0477 1 ENT, 'Node ID (node-name, node-address): ',  
: 485      P 0478 1 :  
: 486      0479 1 :  
: 487      0480 1 :  
: 488      0481 1 :  
: 489      P 0482 1 PROMPT_STRINGS  
: 490      (CLI,  
: 491      P 0484 1 :  
: 492      P 0485 1 ALL, 'All line parameters (Y, N): ',  
: 493      P 0486 1 ENT, 'Line ID (dev-c-u.t): ',  
: 494      P 0487 1 CTM, 'Counter timer (Y, N): ',  
: 495      P 0488 1 DDT, 'Dead timer (Y, N): ',  
: 496      P 0489 1 DLT, 'Delay timer (Y, N): ',  
: 497      P 0490 1 HTI, 'Holdback timer (Y, N): ',  
: 498      P 0491 1 MRT, 'Maximum retransmits (Y, N): ',  
: 499      P 0492 1 SLT, 'Scheduling timer (Y, N): ',  
: 500      P 0493 1 SRT, 'Stream timer (Y, N): ',  
: 501      P 0494 1 :  
: 502      0495 2 )  
: 503      0496 1 :  
: 504      0497 1 :  
: 505      P 0498 1 PROMPT_STRINGS  
: 506      (CCI,  
: 507      P 0500 1 :  
: 508      P 0501 1 ALL, 'All circuit parameters (Y, N): ',  
: 509      P 0502 1 ENT, 'Circuit ID (16 characters): ',  
: 510      P 0503 1 BBT, 'Babble timer (Y, N): ',  
: 511      P 0504 1 CTM, 'Counter timer (Y, N): ',  
: 512      P 0505 1 MRB, 'Maximum receive buffers (Y, N): ',  
: 513      P 0506 1 !!! MRC, 'Maximum recalls (Y, N): ',
```

: 514 P 0507 1 OWN, 'Owner (Y, N): '
: 515 P 0508 1 RCT, 'Recall timer (Y, N): '
: 516 P 0509 1 TRT, 'Transmit timer (Y, N): '
: 517 P 0510 1 ACB, 'Active base (Y, N): '
: 518 P 0511 1 ACI, 'Active increment (Y, N): '
: 519 P 0512 1 IAB, 'Inactive base (Y, N): '
: 520 P 0513 1 IAI, 'Inactive increment (Y, N): '
: 521 P 0514 1 IAT, 'Inactive threshold (Y, N): '
: 522 P 0515 1 DYB, 'Dying base (Y, N): '
: 523 P 0516 1 DYI, 'Dying increment (Y, N): '
: 524 P 0517 1 DYT, 'Dying threshold (Y, N): '
: 525 P 0518 1 DTH, 'Dead threshold (Y, N): '
: 526 P 0519 1
: 527 0520 2)
: 528 0521 1 .
: 529 0522 1
: 530 P 0523 1 PROMPT_STRINGS
: 531 P 0524 1 (CLO,
: 532 P 0525 1
: 533 P 0526 1 ALL, 'All logging of the specified type (Y, N): '
: 534 P 0527 1 ENT, 'Type of logging (CONSOLE, FILE, MONITOR): '
: 535 P 0528 1 EVK, 'Event list (cls.typ-typ,typ, KNOWN): '
: 536 P 0529 1 NAM, 'Name of sink file (Y, N): '
: 537 L 0530 1 SRC, %STRING('Source of events (LINE line-id, ', CRLF,
: 538 P 0531 1 (CIRCUIT circ-id, NODE node-id):),
: 539 P 0532 1 SNO, 'Sink node (node-id, EXECUTOR): '
: 540 P 0533 1
: 541 0534 2)
: 542 0535 1 :

```
: 544      0536 1
: 545      0537 1 BIND
: 546      0538 1
: 547      P 0539 1 PROMPT_STRINGS
: 548      P 0540 1 (COB,
: 549      P 0541 1
: 550      P 0542 1 ALL, 'All parameters (Y, N): '
: 551      P 0543 1 ENT, 'Object name (object-name): '
: 552      P 0544 1 FIL, 'File name (Y, N): '
: 553      P 0545 1 NUM, 'Object number (Y, N): '
: 554      P 0546 1 PRV, 'Privileges (Y, N): '
: 555      P 0547 1 USR, 'User ID (Y, N): '
: 556      P 0548 1 PSW, 'Password (Y, N): '
: 557      P 0549 1 ACC, 'Account (Y, N): '
: 558      P 0550 1 PRX, 'Proxy access (Y, N): '
: 559      P 0551 1
: 560      0552 2 )
: 561      0553 1 :
```

```
: 563    0554 1 %SBTTL 'State Table Entry'  
: 564    0555 1  
: 565    0556 1 $INIT_STATE (NCP$G_STTBL_CLPU, NCP$G_KYTBL_CLPU);  
: 566    0557 1  
: 567    0558 1  
: 568    0559 1 |  
: 569    0560 1 | CLEAR/PURGE Commands  
: 570    0561 1 |  
: 571    0562 1 |  
: 572    0563 1 | Dispatch to entity types and prompt if end of string  
: 573    0564 1 |  
: 574    0565 1 |  
: 575    P 0566 1 | COMMAND PROMPT  
: 576    P 0567 1 | (CLPU, ENT, NCPS_INVKEY,  
: 577    P 0568 1 |  
: 578    P 0569 1 | ('CIRCUIT', ST_CCI_ENT, NMASC_ENT_CIR, NCPSGL_OPTION, ),  
: 579    P 0570 1 | ('CONFIGURATOR', ST_CLPO_CNF),  
: 580    P 0571 1 | ('CONSOLE', ST_CLPU_CNS),  
: 581    P 0572 1 | ('DTE', ST_CLPU_CPRDTE),  
: 582    P 0573 1 | ('EXECUTOR', TPAS_EXIT, ACTSNXT STATE,  
: 583    P 0574 1 |     NMASC_ENT_NOD, NCPSGL_OPTION, NEXT_STATE(CEX)),  
: 584    P 0575 1 | ('GROUP', ST_CLPU_CPRGRP),  
: 585    P 0576 1 | ('LINE', ST_CLI_ENT, NMASC_ENT_LIN, NCPSGL_OPTION, ),  
: 586    P 0577 1 | ('LOADER', ST_CLPO_LOA),  
: 587    P 0578 1 | ('LOOPER', ST_CLPU_LOO),  
: 588    P 0579 1 | ('LOGGING', ST_CLO_ENT, NMASC_ENT_LOG, NCPSGL_OPTION, ),  
: 589    P 0580 1 | ('KNOWN', ST_CLPO_KWN, ACTSSAVPRM, , PBKSG_CLPU_KWN),  
: 590    P 0581 1 | ('MODULE', ST_CLPU_MOD),  
: 591    P 0582 1 | ('NODE', ST_CNO_ENT, NMASC_ENT_NOD, NCPSGL_OPTION, ),  
: 592    P 0583 1 | ('OBJECT', ST_COB_ENT, NMASC_SENT_OBJ, NCPSGE_OPTION, ),  
: 593    P 0584 1 | ('TRACEPOINT', ST_CLPO_CTRPT),  
: 594    P 0585 1 | ('X25', ST_CLPU_X25),  
: 595    P 0586 1 | ('X29', ST_CLPU_X29)  
: 596    0587 1 )
```

```
: 598      0588 1 %SBTTL 'Clear/Purge Known'  
: 599      0589 1  
: 600      0590 1  
: 601      0591 1      Clear purge known  
: 602      0592 1  
: 603      0593 1  
: 604      0594 1  
: 605      0595 1      Dispatch for the entity types, prompt if none  
: 606      0596 1  
: 607      0597 1  
P 608      0598 1      COMMAND PROMPT  
: 609      0599 1      (CLPU, RWN, NCPS_INVKEY,  
: 610      0600 1  
: 611      0601 1      ('CIRCUITS', ST_CCI_GO, , NMASC_ENT_CIR, NCPSGL_OPTION, ),  
: 612      0602 1      ('LINES', ST_CLI_GO, , NMASC_ENT_LIN, NCPSGL_OPTION, ),  
: 613      0603 1      ('LOGGING', ST_CLO_GO, , NMASC_ENT_LOG, NCPSGL_OPTION, ),  
: 614      0604 1      ('NODES', TPAS_EXIT, ACT$NXt STATE,  
: 615      0605 1      NMASC_ENT_NOD, NCPSGE_OPCIÓN, NEXT STATE(CNO)),  
: 616      0606 1      ('OBJECTS', ST_COB_GO, , NMASC_SENT_OBJ, NCPSGL_OPCIÓN, )  
: 617      0607 1  
: 618      0608 1      )
```

```
; 620      0609 1 %SBTTL 'Clear/Purge Module'  
; 621      0610 1  
; 622      0611 1 !  
; 623      0612 1 : Modules  
; 624      0613 1 :  
; 625      0614 1 :  
; 626      P 0615 1 COMMAND PROMPT  
; 627      (CLPU, MOD, NCPS_INVKEY,  
; 628      P 0616 1  
; 629      P 0617 1  
; 630      P 0618 1 ('CONFIGURATOR', ST_CLPU_CNF),  
; 631      P 0619 1 ('CONSOLE', ST_CLPU_CNS),  
; 632      P 0620 1 ('LOADER', ST_CLPU_LOA),  
; 633      P 0621 1 ('LOOPER', ST_CLPU_LOO),  
; 634      P 0622 1 ('X25', ST_CLPU_X25),  
; 635      P 0623 1 ('X29', ST_CLPU_X29)  
; 636      P 0624 1 )  
; 637      P 0625 1 $STATE (ST_CLPU_X25,  
; 638      0626 1 ('-'));  
; 639      P 0627 1 $STATE {  
; 640      P 0628 1 ('ACCESS', ST_CLPU_CAC),  
; 641      P 0629 1 ('PROTOCOL', ST_CLPU_CPR),  
; 642      P 0630 1 ('SERVER', ST_CLPU_CSE),  
; 643      P 0631 1 ('TRACE', ST_CLPU_CTR)  
; 644      P 0632 1 );  
; 645      P 0633 1 :  
; 646      P 0634 1 $STATE (ST_CLPU_X29,  
; 647      0635 1 ('-'));  
; 648      P 0636 1 $STATE {  
; 649      P 0637 1 ('SERVER', ST_CLPU_C9S)  
; 650      P 0638 1 );  
; 651      P 0639 1 :  
; 652      P 0640 1 :  
; 653      P 0641 1 $STATE (ST_CLPU_CNF, ! Module Configurator  
; 654      (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT STATE,  
; 655      NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CCF))  
; 656      P 0642 1 :  
; 657      P 0643 1 :  
; 658      P 0644 1 $STATE (ST_CLPU_CNS, ! Module Console  
; 659      (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT STATE,  
; 660      NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CCS))  
; 661      P 0645 1 :  
; 662      P 0646 1 :  
; 663      P 0647 1 :  
; 664      P 0648 1 :  
; 665      P 0649 1 :  
; 666      P 0650 1 :  
; 667      P 0651 1 :  
; 668      P 0652 1 $STATE (ST_CLPU_LOA, ! Module Loader  
; 669      (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT STATE,  
; 670      NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CLD))  
; 671      P 0653 1 :  
; 672      P 0654 1 :  
; 673      P 0655 1 :  
; 674      P 0656 1 :  
; 675      P 0657 1 $STATE (ST_CLPU_LOO, ! Module looper  
; 676      (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT STATE,  
; 677      NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CLW))  
; 678      P 0658 1 :  
; 679      P 0659 1 :  
; 680      P 0660 1 :  
; 681      P 0661 1 :  
; 682      P 0662 1 $STATE (ST_CLPU_CAC, ! X25-Access  
; 683      (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT STATE,  
; 684      NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CAC))  
; 685      P 0663 1 :  
; 686      P 0664 1 :  
; 687      P 0665 1 :  
;
```

677 0666 1
678 P 0667 1 \$STATE (ST_CLPU_CPR,
679 P 0668 1 (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT_STATE,
680 P 0669 1 NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CPR))
681 0670 1); ! X25-Protocol
682 P 0671 1 \$STATE (ST_CLPU_CPRDTE,
683 P 0672 1 (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT_STATE,
684 P 0673 1 NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CPRDTE))
685 P 0674 1); ! X25-Protocol DTE
686 P 0675 1 \$STATE (ST_CLPU_CPRGRP,
687 P 0676 1 (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT_STATE,
688 P 0677 1 NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CPRGRP))
689 0678 1); ! X25-Protocol Group
690 0679 1
691 P 0680 1 \$STATE (ST_CLPU_CSE,
692 P 0681 1 (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT_STATE,
693 P 0682 1 NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CSE))
694 0683 1);
695 0684 1
696 P 0685 1 \$STATE (ST_CLPU_CTR,
697 P 0686 1 (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT_STATE,
698 P 0687 1 NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CTR))
699 0688 1); ! X25-Trace
700 P 0689 1 \$STATE (ST_CLPU_CTRPT,
701 P 0690 1 (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT_STATE,
702 P 0691 1 NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(CTRPT))
703 0692 1); ! X25-Trace Tracepoint
704 0693 1
705 P 0694 1 \$STATE (ST_CLPU_C9S,
706 P 0695 1 (TPAS_LAMBDA, TPAS_EXIT, ACTSNXT_STATE,
707 P 0696 1 NMASC_ENT_MOD, NCPSGL_OPTION, NEXT_STATE(C9S))
708 0697 1); ! X29-Server

; 710 0698 1 %SBTTL 'Clear/Purge Node'
; 711 0699 1
; 712 0700 1
; 713 0701 1 Clear purge Node
; 714 0702 1
; 715 0703 1
; 716 0704 1
; 717 0705 1 Collect the node id or prompt
; 718 0706 1
; 719 0707 1
; 720 P 0708 1 COMMAND PROMPT
; 721 P 0709 1 (CNO, ENT, NCPS_INVVAL,
; 722 P 0710 1
; 723 P 0711 1 ((SE_NODE_ID), , ACT\$SAVPRM, , , PBK\$G_CNO_ENT)
; 724 P 0712 1
; 725 P 0713 1)
; 726 P 0714 1
; 727 P 0715 1 \$STATE (
; 728 P 0716 1 (TPAS_LAMBDA, TPAS_EXIT, ACT\$NXT_STATE,
; 729 P 0717 1 NMASC_ENT_NOD, NCPSGL_OPTION, NEXT_STATE(CNO))
; 730 P 0718 1):
; 731 P 0719 1

```
; 733      0720 1 %SBTTL 'Clear/Purge Circuits'  
; 734      0721 1  
; 735      0722 1  
; 736      0723 1 | Clear purge circuits  
; 737      0724 1 |  
; 738      0725 1 |  
; 739      0726 1 |  
; 740      0727 1 | Collect the circuit id and save it, prompt if none  
; 741      0728 1 |  
; 742      0729 1 |  
; 743      P 0730 1 | COMMAND PROMPT  
; 744      P 0731 1 | (CCI, ENT, NCPS_INVVAL,  
; 745      P 0732 1 |  
; 746      P 0733 1 | ( (SE_CIRC_ID), , ACT$SAVPRM, , , PBK$G_CCI_ENT)  
; 747      P 0734 1 |  
; 748      0735 1 | )  
; 749      0736 1 |  
; 750      0737 1 |  
; 751      0738 1 | Come here after KNOWN and either prompt  
; 752      0739 1 | or dispatch to process args  
; 753      0740 1 |  
; 754      0741 1 |  
; 755      P 0742 1 $STATE (ST_CCI_GO,  
; 756      P 0743 1 (TPAS_EOS),  
; 757      P 0744 1 (TPAS_LAMBDA, ST_CCI_PRC)  
; 758      0745 1 );  
; 759      0746 1  
; 760      P 0747 1 QUERY_STATES ! Build states to query for params  
; 761      P 0748 1 (CCI,  
; 762      P 0749 1  
; 763      P 0750 1 ALL  
; 764      P 0751 1  
; 765      0752 1 )  
; 766      0753 1  
; 767      0754 1 |  
; 768      0755 1 | Perform the function if there is no junk on the line  
; 769      0756 1 |  
; 770      0757 1 |  
; 771      P 0758 1 $STATE (ST_CCI_DOIT, ! Set the option mask  
; 772      P 0759 1 (TPAS_LAMBDA, , NMASC_ENT_CIR, NCPSGL_OPTION, )  
; 773      0760 1 );  
; 774      0761 1  
; 775      P 0762 1 $STATE ( ! Perform the function  
; 776      P 0763 1 (TPAS_EOS, TPAS_EXIT, ACT$VRB.Utility, , SDB$G_CCI)  
; 777      0764 1 );
```

```
: 779      0765 1
: 780      0766 1
: 781      0767 1 : Dispatch for the parameters
: 782      0768 1
: 783      0769 1
: 784      P 0770 1 $STATE (ST_CCI_PRC,
: 785      P 0771 1
: 786      P 0772 1 DISPATCH_STATES
: 787      P 0773 1 (CCI,
: 788      P 0774 1
: 789      P 0775 1 ACT, 'ACTIVE'.
: 790      P 0776 1 ALL, 'ALL'
: 791      P 0777 1 BBT, 'BABBLE'
: 792      P 0778 1 BLK, 'BLOCKING'
: 793      P 0779 1 CHN, 'CHANNEL'
: 794      P 0780 1 COS, 'COST'
: 795      P 0781 1 CTM, 'COUNTER'
: 796      P 0782 1 DTH, 'DEAD'
: 797      P 0783 1 DTE, 'DTE'
: 798      P 0784 1 DYE, 'DYING'
: 799      P 0785 1 HET, 'HELLO'
: 800      P 0786 1 IAC, 'INACTIVE'
: 801      P 0787 1 LIN, 'LINE'
: 802      P 0788 1 LIT, 'LISTEN'
: 803      P 0789 1 MAX, 'MAXIMUM'
: 804      P 0790 1 NUM, 'NUMBER'
: 805      P 0791 1 OWN, 'OWNER'
: 806      P 0792 1 POL, 'POLLING'
: 807      P 0793 1 RCT, 'RECALL'
: 808      P 0794 1 RPR, 'ROUTER'
: 809      P 0795 1 SER, 'SERVICE'
: 810      P 0796 1 STA, 'STATE'
: 811      P 0797 1 TRT, 'TRANSMIT'
: 812      P 0798 1 XPT, 'TRANSPORT'
: 813      P 0799 1 TRI, 'TRIBUTARY'
: 814      P 0800 1 TYP, 'TYPE'
: 815      P 0801 1 USE, 'USAGE'
: 816      P 0802 1 VER, 'VERIFICATION'
: 817      P 0803 1 )
: 818      P 0804 1
: 819      P 0805 1
: 820      P 0806 1 (TPAS_EOS, ST_CCI_DOIT)
: 821      0807 1 );
```

: 823 0808 1
.: 824 0809 1
.: 825 0810 1 These states take care of noise words
.: 826 0811 1 and call the subexpressions to do the work
.: 827 0812 1
.: 828 0813 1
.: 829 P 0814 1 PROCESS_STATES
.: 830 P 0815 1 (CCI,
.: 831 P 0816 1
.: 832 P 0817 1 ALL, .
.: 833 P 0818 1 STA, .
.: 834 P 0819 1 SER, .
.: 835 P 0820 1 CTM, 'TIMER',
.: 836 P 0821 1 COS, .
.: 837 P 0822 1 MRT, . ! From MAX below
.: 838 P 0823 1 RPR, 'PRIORITY',
.: 839 P 0824 1 HET, 'TIMER',
.: 840 P 0825 1 LIT, 'TIMER',
.: 841 P 0826 1 BLK, .
.: 842 P 0827 1 MRC, . ! From MAX below
.: 843 P 0828 1 RCT, 'TIMER',
.: 844 P 0829 1 NUM, .
.: 845 P 0830 1 POL, 'STATE',
.: 846 P 0831 1 OWN, .
.: 847 P 0832 1 LIN, .
.: 848 P 0833 1 USE, .
.: 849 P 0834 1 TYP, .
.: 850 P 0835 1 DTE, .
.: 851 P 0836 1 CHN, .
.: 852 P 0837 1 MBL, . ! From MAX below
.: 853 P 0838 1 MWI, . ! From MAX below
.: 854 P 0839 1 TRI, .
.: 855 P 0840 1 BBT, 'TIMER',
.: 856 P 0841 1 TRT, 'TIMER',
.: 857 P 0842 1 MRB, . ! From MAX below
.: 858 P 0843 1 MTR, . ! From MAX below
.: 859 P 0844 1 ACB, . ! From ACT below
.: 860 P 0845 1 ACI, . ! From ACT below
.: 861 P 0846 1 IAB, . ! From IAC below
.: 862 P 0847 1 IAI, . ! From IAC below
.: 863 P 0848 1 IAT, . ! From IAC below
.: 864 P 0849 1 DYB, . ! From DYE below
.: 865 P 0850 1 DYI, . ! From DYE below
.: 866 P 0851 1 DYT, . ! From DYE below
.: 867 P 0852 1 DTH, 'THRESHOLD',
.: 868 P 0853 1 VER,
.: 869 P 0854 1 XPT, 'TYPE',
.: 870 P 0855 1)
.: 871 P 0856 1
.: 872 0857 1

: 874 0858 1
: 875 0859 1
: 876 0860 1
: 877 0861 1
: 878 0862 1
: 879 P 0863 1 SUB EXPRESSIONS
: 880 P 0864 1 (CCI.
: 881 P 0865 1
: 882 P 0866 1 ALL, TPAS_EOS.
: 883 P 0867 1
: 884 P 0868 1 STA, TPAS_LAMBDA.
: 885 P 0869 1 SCR, TPAS_LAMBDA.
: 886 P 0870 1 C(M, TPAS_LAMBDA.
: 887 P 0871 1 COS, TPAS_LAMBDA.
: 888 P 0872 1 MRT, TPAS_LAMBDA.
: 889 P 0873 1 RPR, TPAS_LAMBDA.
: 890 P 0874 1 MET, TPAS_LAMBDA.
: 891 P 0875 1 LIT, TPAS_LAMBDA.
: 892 P 0876 1 BLK, TPAS_LAMBDA.
: 893 P 0877 1 MRC, TPAS_LAMBDA.
: 894 P 0878 1 RCT, TPAS_LAMBDA.
: 895 P 0879 1 NUM, TPAS_LAMBDA.
: 896 P 0880 1 POL, TPAS_LAMBDA.
: 897 P 0881 1 OWN, TPAS_LAMBDA.
: 898 P 0882 1 LIN, TPAS_LAMBDA.
: 899 P 0883 1 USE, TPAS_LAMBDA.
: 900 P 0884 1 TYP, TPAS_LAMBDA.
: 901 P 0885 1 DTE, TPAS_LAMBDA.
: 902 P 0886 1 CHN, TPAS_LAMBDA.
: 903 P 0887 1 MBL, TPAS_LAMBDA.
: 904 P 0 38 1 MWI, TPAS_LAMBDA.
: 905 P 0889 1 TRI, TPAS_LAMBDA.
: 906 P 0890 1 BBT, TPAS_LAMBDA.
: 907 P 0891 1 TRT, TPAS_LAMBDA.
: 908 P 0892 1 MRB, TPAS_LAMBDA.
: 909 P 0893 1 MTR, TPAS_LAMBDA.
: 910 P 0894 1 ACB, TPAS_LAMBDA.
: 911 P 0895 1 ACI, TPAS_LAMBDA.
: 912 P 0896 1 IAB, TPAS_LAMBDA.
: 913 P 0897 1 IAI, TPAS_LAMBDA.
: 914 P 0898 1 IAT, TPAS_LAMBDA.
: 915 P 0899 1 DYB, TPAS_LAMBDA.
: 916 P 0900 1 DYI, TPAS_LAMBDA.
: 917 P 0901 1 DYT, TPAS_LAMBDA.
: 918 P 0902 1 DTH, TPAS_LAMBDA.
: 919 P 0903 1
: 920 P 0904 1 VER, TPAS_LAMBDA.
: 921 P 0905 1 XPT, TPAS_LAMBDA.
: 922 P 0906 1
: 923 0907 1)

These subexpressions store away the parameter id's

925 0908 1
926 0909 1 |
927 0910 1 | Dispatch on MAXIMUM keyword
928 0911 1 |
929 0912 1 |
930 P 0913 1 \$STATE (ST_CCI_PRC_MAX,
931 P 0914 1
932 P 0915 1 DISPATCH_STATES
933 P 0916 1 (CCI,
934 P 0917 1
935 P 0918 1 MBL, 'DATA'
936 P 0919 1 MRC, 'RECALLS'
937 P 0920 1 MRT, 'ROUTERS'
938 P 0921 1 MTR, 'TRANSMITS'
939 P 0922 1 MWI, 'WINDOW'
940 P 0923 1
941 P 0924 1)
942 0925 1);
943 0926 1
944 0927 1 |
945 0928 1 | Dispatch on ACTIVE keyword
946 0929 1 |
947 0930 1 |
948 P 0931 1 \$STATE (ST_CCI_PRC_ACT,
949 P 0932 1
950 P 0933 1 DISPATCH_STATES
951 P 0934 1 (CCI,
952 P 0935 1
953 P 0936 1 ACB, 'BASE'
954 P 0937 1 ACI, 'INCREMENT'
955 P 0938 1
956 P 0939 1)
957 0940 1);
958 0941 1
959 0942 1 |
960 0943 1 | Dispatch for INACTIVE keyword
961 0944 1 |
962 0945 1 |
963 P 0946 1 \$STATE (ST_CCI_PRC_IAC, ! Inactive things
964 P 0947 1
965 P 0948 1 DISPATCH_STATES
966 P 0949 1 (CCI,
967 P 0950 1
968 P 0951 1 IAB, 'BASE'
969 P 0952 1 IAI, 'INCREMENT'
970 P 0953 1 IAT, 'THRESHOLD';
971 P 0954 1
972 P 0955 1)
973 0956 1);
974 0957 1
975 0958 1 |
976 0959 1 | Dispatch for DYING keyword
977 0960 1 |
978 0961 1 |
979 P 0962 1 \$STATE (ST_CCI_PRC_DYE,
980 P 0963 1 DISPATCH_STATES
981 P 0964 1

NCPSTACLP
V04-000

Clear/Purge Parse States and Data
Clear/Purge Circuits

M 10
16-Sep-1984 00:05:04
14-Sep-1984 12:48:17

VAX-11 Bliss-32 V4.0-742
[NCP.SRC]NCPSTACLP.B32;1

Page 25
(18)

: 982 P 0965 1 (CCI,
: 983 P 0966 1
: 984 P 0967 1 DYB. 'BASE'
: 985 P 0968 1 DYI. 'INCREMENT'
: 986 P 0969 1 DYT. 'THRESHOLD'
: 987 P 0970 1
: 988 P 0971 1)
: 989 P 0972 1);

NC
VO

```
; 991      0973 1 %SBTTL 'Clear/Purge Lines'  
; 992      0974 1  
; 993      0975 1  
; 994      0976 1      Clear purge lines  
; 995      0977 1  
; 996      0978 1  
; 997      0979 1  
; 998      0980 1      Collect the line id and save it, prompt if none  
; 999      0981 1  
; 1000     0982 1  
; 1001     P 0983 1      COMMAND PROMPT  
; 1002     P 0984 1      (CLI, ENT, NCPS_INVAL,  
; 1003     P 0985 1  
; 1004     P 0986 1      ( (SE_LINE_ID), , ACTSSAVPRM, , , PBK$G_CLI_ENT)  
; 1005     P 0987 1  
; 1006     0988 1  
; 1007     0989 1  
; 1008     0990 1  
; 1009     0991 1      Come here after KNOWN and either prompt  
; 1010     0992 1      or dispatch to process args  
; 1011     0993 1  
; 1012     0994 1  
; 1013     P 0995 1 $STATE (ST_CLI_GO,  
; 1014     P 0996 1      (TPAS_EOS),  
; 1015     P 0997 1      (TPAS_LAMBDA, ST_CLI_PRC)  
; 1016     0998 1  
; 1017     0999 1  
; 1018     P 1000 1      QUERY_STATES           ! Build states to query for params  
; 1019     P 1001 1      (CLI,  
; 1020     P 1002 1  
; 1021     P 1003 1      ALL  
; 1022     P 1004 1  
; 1023     1005 1  
; 1024     1006 1  
; 1025     1007 1  
; 1026     1008 1      Perform the function if there is no junk on the line  
; 1027     1009 1  
; 1028     1010 1  
; 1029     P 1011 1 $STATE (ST_CLI_DOIT,           ! Set the option mask  
; 1030     P 1012 1      (TPAS_LAMBDA, , NMASC_ENT_LIN, NCPSGL_OPTION, )  
; 1031     1013 1  
; 1032     1014 1  
; 1033     P 1015 1 $STATE (,           ! Perform the function  
; 1034     P 1016 1      (TPAS_EOS, TPAS_EXIT, ACTSVRB.Utility, , SDB$G_CLI)  
; 1035     1017 1
```

```
: 1037      1018 1
: 1038      1019 1 :
: 1039      1020 1 : Dispatch for line parameters
: 1040      1021 1 :
: 1041      1022 1
: 1042      P 1023 1 $STATE (ST_CLI_PRC,
: 1043      PP 1024 1
: 1044      PP 1025 1 DISPATCH_STATES
: 1045      PP 1026 1 (CLI,
: 1046      PP 1027 1
: 1047      PP 1028 1 ALL, 'ALL',
: 1048      PP 1029 1 CLO, 'CLOCK',
: 1049      PP 1030 1 CON, 'CONTROLLER',
: 1050      PP 1031 1 COS, 'COST'
: 1051      PP 1032 1 CTM, 'COUNTER',
: 1052      PP 1033 1 DDT, 'DEAD'
: 1053      PP 1034 1 DLT, 'DELAY'
: 1054      PP 1035 1 DEV, 'DEVICE'
: 1055      PP 1036 1 DPX, 'DUPLEX'
: 1056      PP 1037 1 HTI, 'HOLDBACK'
: 1057      PP 1038 1 MAX, 'MAXIMUM'
: 1058      PP 1039 1 MCD, 'MICROCODE'
: 1059      PP 1040 1 XMD, 'MODE'
: 1060      PP 1041 1 NTM, 'RETRANSMIT'
: 1061      PP 1042 1 NTM, 'NORMAL'          ! [V2 only]
: 1062      PP 1043 1 PRO, 'PROTOCOL'
: 1063      PP 1044 1 BFN, 'RECEIVE'
: 1064      PP 1045 1 SLT, 'SCHEDULING'
: 1065      PP 1046 1 SVC, 'SERVICE'
: 1066      PP 1047 1 STA, 'STATE'
: 1067      PP 1048 1 SRT, 'STREAM'
: 1068      PP 1049 1 TRB, 'TRIBUTARY'
: 1069      PP 1050 1 TYP, 'TYPE'
: 1070      PP 1051 1
: 1071      PP 1052 1 )
: 1072      PP 1053 1
: 1073      P 1054 1 ;(TPAS_EOS, ST_CLI_DOIT)
: 1074      1055 1 ;
```

```
: 1076    1056 1
: 1077    1057 1
: 1078    1058 1 | These states take care of noise words
: 1079    1059 1 | and call the subexpressions to do the work
: 1080    1060 1
: 1081    1061 1
P 1082    1062 1 PROCESS_STATES
P 1083    1063 1 (CLI,
P 1084    1064 1
P 1085    1065 1 ALL, .
P 1086    1066 1 CLO, .
P 1087    1067 1 CON, .
P 1088    1068 1 COS, .
P 1089    1069 1 CTM, 'TIMER',
P 1090    1070 1 DDT, 'TIMER',
P 1091    1071 1 DLT, 'TIMER',
P 1092    1072 1 DEV, .
P 1093    1073 1 DPX,
P 1094    1074 1 HTI, 'TIMER',
P 1095    1075 1 MCD, 'DUMP',
P 1096    1076 1 MWI, .           ! From MAX below
P 1097    1077 1 MRT, .           ! From MAX below
P 1098    1078 1 MBL, .           ! From MAX below
P 1099    1079 1 NTM, 'TIMER',
P 1100    1080 1 PRO,
P 1101    1081 1 BFN, 'BUFFERS',
P 1102    1082 1 SLT, 'TIMER',
P 1103    1083 1 STM, .
P 1104    1084 1 SVM, .
P 1105    1085 1 STA,
P 1106    1086 1 SRT, 'TIMER',
P 1107    1087 1 TRB, .
P 1108    1088 1 TYP, .
P 1109    1089 1 XMD, .
P 1110    1090 1
P 1111    1091 1 )
```

: 1113 1092 1
: 1114 1093 1
: 1115 1094 1
: 1116 1095 1
: 1117 1096 1
: 1118 P 1097 1
: 1119 P 1098 1
: 1120 P 1099 1
: 1121 P 1100 1
: 1122 P 1101 1
: 1123 P 1102 1
: 1124 P 1103 1
: 1125 P 1104 1
: 1126 P 1105 1
: 1127 P 1106 1
: 1128 P 1107 1
: 1129 P 1108 1
: 1130 P 1109 1
: 1131 P 1110 1
: 1132 P 1111 1
: 1133 P 1112 1
: 1134 P 1113 1
: 1135 P 1114 1
: 1136 P 1115 1
: 1137 P 1116 1
: 1138 P 1117 1
: 1139 P 1118 1
: 1140 P 1119 1
: 1141 P 1120 1
: 1142 P 1121 1
: 1143 P 1122 1
: 1144 P 1123 1
: 1145 P 1124 1
: 1146 P 1125 1
: 1147 P 1126 1
: 1148 1127 1

These subexpressions store away the parameter id's

SUB EXPRESSIONS
(CLI,

ALL, TPAS_EOS,

CLO, TPAS_LAMBDA.
CON, TPAS_LAMBDA.
CCS, TPAS_LAMBDA.
CTM, TPAS_LAMBDA.
DDT, TPAS_LAMBDA.
DLT, TPAS_LAMBDA.
DEV, TPAS_LAMBDA.
DPX, TPAS_LAMBDA.
HTI, TPAS_LAMBDA.
MBL, TPAS_LAMBDA.
MCD, TPAS_LAMBDA.
MRT, TPAS_LAMBDA.
MWI, TPAS_LAMBDA.
NTM, TPAS_LAMBDA.
PRO, TPAS_LAMBDA.
BFN, TPAS_LAMBDA.
SLT, TPAS_LAMBDA.
STM, TPAS_LAMBDA.
SVM, TPAS_LAMBDA.
STA, TPAS_LAMBDA.
SRT, TPAS_LAMBDA.
TRB, TPAS_LAMBDA.
TYP, TPAS_LAMBDA.
XMD, TPAS_LAMBDA.

)

```
: 1150      1128 1 !
: 1151      1129 1 ! Dispatch on MAXIMUM keyword
: 1152      1130 1 !
: 1153      1131 1 !
: 1154      P 1132 1 $STATE (ST_CLI_PRC_MAX,
: 1155      P 1133 1
: 1156      P 1134 1 DISPATCH_STATES
: 1157      (CLI,
: 1158      P 1136 1
: 1159      P 1137 1 MBL: 'BLOCK'
: 1160      P 1138 1 MRT: 'RETRANSMITS',
: 1161      P 1139 1 MWI: 'WINDOW',
: 1162      P 1140 1
: 1163      P 1141 1 )
: 1164      1142 1 );
: 1165      1143 1
: 1166      1144 1 !
: 1167      1145 1 ! Dispatch on line SERVICE keyword
: 1168      1146 1 !
: 1169      1147 1 !
: 1170      P 1148 1 $STATE (ST_CLI_PRC_SVC,
: 1171      P 1149 1
: 1172      P 1150 1 DISPATCH_STATES
: 1173      (CLI,
: 1174      P 1152 1
: 1175      P 1153 1 SVM: 'MODE'
: 1176      P 1154 1 STM: 'TIMER',
: 1177      P 1155 1
: 1178      P 1156 1 )
: 1179      1157 1 );
```

```
: 1181      1158 1 %SBTTL 'Clear/Purge Logging'  
: 1182      1159 1 !  
: 1183      1160 1 !  
: 1184      1161 1 !     Clear purge logging  
: 1185      1162 1 !  
: 1186      1163 1 !  
: 1187      P 1164 1 COMMAND PROMPT  
: 1188      P 1165 1 (CLO, ENT, NCPS_INVVAL,  
: 1189      P 1166 1  
: 1190      P 1167 1 ( (SE_LOG_TYP) )           ! Collect the logging type or prompt  
: 1191      P 1168 1 )  
: 1192      1169 1 )  
: 1193      1170 1 ! Come here from elsewhere  
: 1194      P 1171 1 $STATE (ST_CLO_GO,  
: 1195      P 1172 1 (TPAS_LAMBDA, , ACTSSAVPRM, , , PBK$G_EVE_ESET)  
: 1196      1173 1 );  
: 1197      1174 1  
: 1198      P 1175 1 $STATE (,  
: 1199      P 1176 1 (TPAS_EOS),  
: 1200      P 1177 1 (TPAS_LAMBDA, ST_CLO_PRC)    ! And prompt or  
: 1201      1178 1 );                         ! Process command line  
: 1202      1179 1  
: 1203      P 1180 1 QUERY_STATES  
: 1204      P 1181 1 (CLO,  
: 1205      P 1182 1 ALL                      ! All is special, must query  
: 1206      P 1183 1  
: 1207      P 1184 1 )  
: 1208      1185 1  
: 1209      1186 1  
: 1210      P 1187 1 PROMPT_STATES  
: 1211      P 1188 1 (CLO,  
: 1212      P 1189 1  
: 1213      P 1190 1 SRC, EVK, SNO          ! Obtain parameters  
: 1214      P 1191 1  
: 1215      1192 1 )  
: 1216      1193 1  
: 1217      P 1194 1 QUERY_STATES  
: 1218      P 1195 1 (CLO,  
: 1219      P 1196 1 NAM                      ! Only a query here  
: 1220      P 1197 1  
: 1221      P 1198 1 )  
: 1222      1199 1 )  
: 1223      1200 1  
: 1224      P 1201 1 $STATE (ST_CLO_DOIT,  
: 1225      P 1202 1 (TPAS_LAMBDA, , NMASC_ENT_LOG, NCP$GL_OPTION, ) ! Process the function  
: 1226      1203 1 );  
: 1227      1204 1  
: 1228      P 1205 1 $STATE (,  
: 1229      P 1206 1 (TPAS_EOS, TPAS_EXIT, ACT$VRB.Utility, , SDB$G_CLO)  
: 1230      1207 1 );  
: 1231      1208 1
```

```
; 1233      1209 1
; 1234      1210 1
; 1235      1211 1      Dispatch States
; 1236      1212 1
; 1237      1213 1
; 1238      P 1214 1 $STATE (ST_CLO_PRC,
; 1239      P 1215 1
; 1240      P 1216 1 DISPATCH_STATES
; 1241      P 1217 1 (CLO,
; 1242      P 1218 1
; 1243      P 1219 1 ALL, 'ALL'
; 1244      P 1220 1 CIR, 'CIRCUIT'
; 1245      P 1221 1 EVL, 'EVENTS'
; 1246      P 1222 1 KNE, 'KNOWN'
; 1247      P 1223 1 LIN, 'LINE'
; 1248      P 1224 1 NOD, 'NODE'
; 1249      P 1225 1 NAM, 'NAME'
; 1250      P 1226 1 SNK, 'SINK'
; 1251      P 1227 1
; 1252      P 1228 1 )
; 1253      P 1229 1 ;(TPAS_EOS, ST_CLO_DOIT) ! Only if end of string
; 1254      1230 1 ;
```

```

: 1256      1231 1
: 1257      1232 1
: 1258      1233 1 | Process states
: 1259      1234 1 |
: 1260      1235 1
: 1261      P 1236 1 PROCESS_STATES
: 1262      P 1237 1 (CLO,
: 1263      P 1238 1
: 1264      P 1239 1 ALL, .
: 1265      P 1240 1 CIR, .
: 1266      P 1241 1 EVL, .
: 1267      P 1242 1 KNE, 'EVENTS'.
: 1268      P 1243 1 LIN, .
: 1269      P 1244 1 NOD, .
: 1270      P 1245 1 NAM, .
: 1271      P 1246 1 SNK, .
: 1272      P 1247 1
: 1273      1248 1 )
: 1274      1249 1
: 1275      P 1250 1 $STATE (ST_CLO_SRC, ! Source must be line, circuit or node
: 1276      P 1251 1 ('CIRCUIT', ST_CLO_CIR),
: 1277      P 1252 1 ('LINE', ST_CLO_LIN),
: 1278      P 1253 1 ('NODE', ST_CLO_NOD),
: 1279      1254 1 );
: 1280      1255 1
: 1281      P 1256 1 $STATE (ST_CLO_SNK, ! Sub-dispatch for sink node or exec
: 1282      P 1257 1 ('EXECUTOR', ST_CLO_SEX),
: 1283      P 1258 1 ('NODE', ST_CLO_SNO),
: 1284      1259 1 );
: 1285      1260 1
: 1286      1261 1
: 1287      1262 1
: 1288      1263 1 | Subexpressions
: 1289      1264 1 |
: 1290      1265 1
: 1291      P 1266 1 SUB_EXPRESSIONS
: 1292      P 1267 1 (CLO,
: 1293      P 1268 1
: 1294      P 1269 1 ALL, TPAS_EOS,
: 1295      P 1270 1 CIR, (SE_TIRC_ID),
: 1296      P 1271 1 EVL, (SE_EVENT_LIST),
: 1297      P 1272 1 KNE, (SE_EVENT_KNOWN),
: 1298      P 1273 1 LIN, (SE_LINE_ID),
: 1299      P 1274 1 NOD, (SE_NODE_ID),
: 1300      P 1275 1 NAM, TPAS_LAMBDA
: 1301      P 1276 1 SNO, (SE_NODE_IDS),
: 1302      P 1277 1 SEX, TPAS_LAMBDA,
: 1303      P 1278 1
: 1304      1279 1 )
: 1305      1280 1
: 1306      1281 1
: 1307      1282 1
: 1308      1283 1 | Subexpression to prompt for events
: 1309      1284 1 |
: 1310      1285 1
: 1311      P 1286 1 $STATE (ST_CLO_EVK,
: 1312      P 1287 1 ( (SE_EVENT_LIST), TPAS_EXIT, ACT$SAVPRM, . , PBK$G_CLO_EVL),

```

```
: 1313 P 1288 1      ('KNOWN', , ACTSSAVPRM, , , PBKSG_CLO_EVL)
: 1314   1289 1      );
: 1315   1290 1
: 1316 P 1291 1 $STATE {,
: 1317 P 1292 1      ('(SE_EVENT_KNOWN), TPAS_EXIT)      ! Store known event code
: 1318   1293 1      );
: 1319   1294 1
```

```
: 1321      1295 1 %SBTTL 'Clear/Purge Objects'  
: 1322      1296 1  
: 1323      1297 1 !  
: 1324      1298 1 | Clear purge objects  
: 1325      1299 1 |  
: 1326      1300 1 |  
: 1327      1301 1 |  
: 1328      1302 1 | Collect the object name/number or prompt  
: 1329      1303 1 |  
: 1330      1304 1 |  
: 1331      P 1305 1 | COMMAND PROMPT  
: 1332      P 1306 1 | (COB, ENT, NCPS_INVVAL,  
: 1333      P 1307 1 |  
: 1334      D 1308 1 | ( (SE_OBJECT_ID), , ACT$SAVPRM, , , PBK$G_COB_ENT)  
: 1335      P 1309 1 |  
: 1336      1310 1 | )  
: 1337      1311 1 |  
: 1338      1312 1 |  
: 1339      1313 1 | Should we prompt or just process the remainder of the command?  
: 1340      1314 1 |  
: 1341      1315 1 |  
: 1342      P 1316 1 $STATE (ST_COB_GO,  
: 1343      P 1317 1 (TPAS_EOS),  
: 1344      P 1318 1 (TPAS_LAMBDA, ST_COB_PRC),  
: 1345      1319 1 );  
: 1346      1320 1  
: 1347      P 1321 1 QUERY_STATES ! Prompt for selected parameters  
: 1348      P 1322 1 (COB,  
: 1349      P 1323 1  
: 1350      P 1324 1 ALL, NUM, FIL, PRV, USR, PSW, PRX  
: 1351      P 1325 1  
: 1352      1326 1 )  
: 1353      1327 1  
: 1354      1328 1  
: 1355      P 1329 1 $STATE (ST_COB_DOIT,  
: 1356      P 1330 1 (TPAS_LAMBDA, , , NMASC_SENT_OBJ, NCPSGL_OPTION, ),  
: 1357      1331 1 );  
: 1358      1332 1  
: 1359      P 1333 1 $STATE (,  
: 1360      P 1334 1 (TPAS_EOS, TPAS_EXIT, ACT$VRB.Utility, , , SDBSG_COB)  
: 1361      1335 1 );
```

```
: 1363      1336 1 %SBTTL 'Dispatch States'  
. 1364      1337 1  
. 1365      1338 1  
. 1366      1339 1 | Dispatch states  
. 1367      1340 1 |  
. 1368      1341 1  
P 1369      1342 1 $STATE (ST_COB_PRC,  
. 1370      1343 1  
P 1371      1344 1 DISPATCH_STATES  
. 1372      1345 1 (COB,  
. 1373      1346 1  
P 1374      1347 1 ALL, 'ALL'  
. 1375      1348 1 ACC, 'ACCOUNT'  
. 1376      1349 1 FIL, 'FILE'  
. 1377      1350 1 NUM, 'NUMBER'  
. 1378      1351 1 PRV, 'PRIVILEGES'  
. 1379      1352 1 PRX, 'PROXY'  
. 1380      1353 1 PSW, 'PASSWORD'  
. 1381      1354 1 USR, 'USER'  
. 1382      1355 1  
P 1383      1356 1 )  
P 1384      1357 1 (TPAS_EOS, ST_COB_DOIT)  
;:  
. 1385      1358 1  
. 1386      1359 1  
P 1387      1360 1 %SBTTL 'Process States'  
. 1388      1361 1  
. 1389      1362 1 |  
. 1390      1363 1 | Process states  
. 1391      1364 1 |  
. 1392      1365 1  
P 1393      1366 1 PROCESS_STATES  
P 1394      1367 1 (COB,  
. 1395      1368 1  
P 1396      1369 1 ALL, .  
. 1397      1370 1 ACC, .  
. 1398      1371 1 FIL, .  
. 1399      1372 1 NUM, .  
. 1400      1373 1 PRV, .  
. 1401      1374 1 PRX, .  
. 1402      1375 1 PSW, .  
. 1403      1376 1 USR, .  
P 1404      1377 1 )  
; 1405      1378 1
```

```
: 1407      1379 1 %SB1TL 'Subexpressions'  
: 1408      1380 1  
: 1409      1381 1 !  
: 1410      1382 1 !       Sub_expressions  
: 1411      1383 1 !  
: 1412      1384 1 !  
: 1413      P 1385 1       SUB_EXPRESSIONS  
: 1414      P 1386 1       (COB.  
: 1415      P 1387 1  
: 1416      P 1388 1       ALL, TPAS_EOS,  
: 1417      P 1389 1       ACC, TPAS_LAMBDA,  
: 1418      P 1390 1       FIL, TPAS_LAMBDA,  
: 1419      P 1391 1       NUM, TPAS_LAMBDA,  
: 1420      P 1392 1       PRV, TPAS_LAMBDA,  
: 1421      P 1393 1       PRX, TPAS_LAMBDA,  
: 1422      P 1394 1       PSW, TPAS_LAMBDA,  
: 1423      P 1395 1       USR, TPAS_LAMBDA,  
: 1424      P 1396 1  
: 1425      1397 1       )
```

NCPSTACLP
V04-000

Clear/Purge Parse States and Data
Define Subexpressions from Library

M 11
16-Sep-1984 00:05:04
14-Sep-1984 12:48:17

VAX-11 Bliss-32 V4.0-742
[NCP.SRC]NCPSTACLP.B32;1

```
: 1427    1398 1 %SBTTL 'Define Subexpressions from Library'  
: 1428    1399 1  
: 1429    1400 1 |  
: 1430    1401 1 | Define subexpressions from Library  
: 1431    1402 1 |  
: 1432    1403 1 |  
: 1433    1404 1 SEM_NODE_ID           | Node id strings  
: 1434    1405 1 SEM_CIRC_ID          | Circuit id strings  
: 1435    1406 1 SEM_LINE_ID           | Line id strings  
: 1436    1407 1 SEM_QUOT_STR          | Quoted strings  
: 1437    1408 1 SEM_OBJECT_ID          | Object name/number  
: 1438    1409 1 SEM_LOG_TYP           | Logging entity type  
: 1439    1410 1 SEM_EVENT_LIST          | List of event numbers  
: 1440    1411 1 SEM_QUERY              | Query state subexpressions
```

NCPSTACLP
V04-000

Clear/Purge Parse States and Data
Object Listing of Parse Table

N 11
16-Sep-1984 00:05:04
14-Sep-1984 12:48:17

VAX-11 Bliss-32 V4.0-742
[NCP.SRC]NCPSTACLP.B32;1

Page 39
(31)

: 1442

1412 1 %SBTTL 'Object Listing of Parse Table'

: 1443

1413 1

: 1444

1414 1 END

: 1445

1415 0 ELUDOM

!End of module

0269 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

